

A Scalable and Distributed Solution to the Inertial Motion Capture Problem

Manon Kok, Sina Khoshfetrat Pakazad, Thomas B. Schön,
Anders Hansson and Jeroen D. Hol

- **Please cite this version:**

Manon Kok, Sina Khoshfetrat Pakazad, Thomas B. Schön, Anders Hansson and Jeroen D. Hol. A Scalable and Distributed Solution to the Inertial Motion Capture Problem. In *Proceedings of the 19th International Conference on Information Fusion*, pp. 1348–1355, Heidelberg, Germany, July 2016.

Abstract

In inertial motion capture, a multitude of body segments are equipped with inertial sensors, consisting of 3D accelerometers and 3D gyroscopes. Using an optimization-based approach to solve the motion capture problem allows for natural inclusion of biomechanical constraints and for modeling the connection of the body segments at the joint locations. The computational complexity of solving this problem grows both with the length of the data set and with the number of sensors and body segments considered. In this work, we present a scalable and distributed solution to this problem using tailored message passing, capable of exploiting the structure that is inherent in the problem. As a proof-of-concept we apply our algorithm to data from a lower body configuration.

A Scalable and Distributed Solution to the Inertial Motion Capture Problem

Manon Kok^{*}, Sina Khoshfetrat Pakazad^{*}, Thomas B. Schön[†],
Anders Hansson^{*} and Jeroen D. Hol[‡]

^{*}Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden

E-mail: {manon.kok, sina.khoshfetratpakazad, anders.g.hansson}@liu.se

[†]Department of Information Technology, Uppsala University, Sweden

E-mail: thomas.schon@it.uu.se

[‡]Xsens Technologies B.V., Enschede, the Netherlands

E-mail: jeroen.hol@xsens.com

August 19, 2016

Abstract

In inertial motion capture, a multitude of body segments are equipped with inertial sensors, consisting of 3D accelerometers and 3D gyroscopes. Using an optimization-based approach to solve the motion capture problem allows for natural inclusion of biomechanical constraints and for modeling the connection of the body segments at the joint locations. The computational complexity of solving this problem grows both with the length of the data set and with the number of sensors and body segments considered. In this work, we present a scalable and distributed solution to this problem using tailored message passing, capable of exploiting the structure that is inherent in the problem. As a proof-of-concept we apply our algorithm to data from a lower body configuration.

1 Introduction

Inertial motion capture focuses on estimating the relative position and orientation (pose) of different human body segments. To this end, inertial sensors (3D accelerometers and 3D gyroscopes) are placed on different body segments as shown in Figure 1. Each body segment’s pose can be estimated by integrating the gyroscope data and double integrating the accelerometer data in time and combining these integrated estimates with a biomechanical model. Inertial sensors are successfully used for full body motion capture in many applications such as character animation, sports and biomechanical analysis [1, 2, 3, 4].

In [5], an optimization-based solution to the inertial motion capture problem was presented. It post-processes the data to obtain a smoothing estimate of the body’s relative pose. The problem is solved using sequential quadratic programming (SQP) [6]. The method was shown to result in drift-free and accurate pose estimates. Using an optimization-based approach allows for natural inclusion of biomechanical constraints and for modeling the connection between the body segment at the joint locations. Furthermore, it naturally handles nonlinearities and opens up the possibility for incorporating non-Gaussian noise and for simultaneous estimation of calibration parameters.

For applications which require real-time pose estimates, approximate solutions to the full smoothing problem need to be considered, for instance using filtering or moving horizon estimation (MHE) [7]. In these approaches, data up to a current time point is used to estimate the current pose. However, in case real-time estimates are not required, all available data can be used to obtain a smoothing estimate. Compared to filtering and MHE, obtaining a smoothing estimate is computationally more expensive and can be challenging both due to the computational complexity of solving the problem and due to storage requirements for constructing the problem. This is specifically of concern when processing long data sets.

In this paper we solve the same problem as in [5]. Again we use SQP, but at each iteration we compute the search directions using the message passing algorithm presented in [8]. This allows us to efficiently make use of the structure inherent in the problem. We exploit this structure in two different ways:



Figure 1: Example of inertial motion capture. Left: olympic and world champion speed skating Ireen Wüst wearing an inertial motion capture suit with 17 inertial sensors. Right: graphical representation of the estimated position and orientation of the body segments.

1. We reorder the problem based on time. This allows us to solve the problem by solving a large number of small problems which enables us to process long data sets.
2. We reorder the problem based on sensors and body segments. This leads to less computational benefits – the number of sensors and body segments is typically much smaller than the number of time steps considered – but it allows for solving the problem in a *distributed* manner. It also relaxes the need for a centralized unit and streaming of data to it.

Using message passing for computing the search directions for the time-ordered problem has close connections to serial dynamic programming [9]. This is due to the chain-like coupling structure in the problem. In fact, using serial dynamic programming, the search directions can be computed by sweeping through the available data forward and backwards, similar to the approach used for Rauch-Tung-Striebel (RTS) smoothing [10]. Using message passing, we compute the search directions by simultaneously starting from the first and final time steps and sweeping towards the middle of the data set and back. This allows us to speed up the search direction computation by a factor of two. Notice that unlike existing scalable algorithms for solving big data problems that rely on first-order methods, see e.g. [11], the proposed algorithm solely relies on second-order methods. Consequently, this algorithm enjoys a far superior superlinear convergence rate, [12], in comparison to at best linear convergence of other algorithms.

If we only consider the lower body for the sensor-ordered problem, the chain-like coupling structure will also be present in the problem. Instead of running through time, this chain runs from one foot through both legs to the other foot. Consequently, it enjoys the same similarities to serial dynamic programming as discussed above. For the full body, the coupling structure will not be chain-like. It will, however, have an inherent tree structure. Hence, we can still use message passing for computing the search directions. In this paper, we focus on the lower body to simplify both the notation and the biomechanical modeling. The presented material can, however, straightforwardly be extended to the full body problem.

The paper is organized as follows. In Section 2 we introduce the inertial motion capture problem for which the models are subsequently introduced in Section 3. In Section 4, we reorder the problem in the two ways described above. These two equivalent formulations of the original problem enjoy a special structure which allows us to use message passing to compute the search directions. The message passing algorithm will be introduced in Section 5. The resulting algorithm that can be used to solve the reordered problems is subsequently discussed in Section 6. In Section 7, we will discuss experimental results where the algorithm is applied to data from inertial sensors placed on the lower body.

2 Problem formulation

The problem of estimating the relative pose of each body segment is formulated as a constrained estimation problem. Given N_T measurements $y = \{y_1, \dots, y_{N_T}\}$, a point estimate of the static parameters θ and the time-varying variables $x = \{x_1, \dots, x_{N_T}\}$ can be obtained as a constrained maximum a posteriori

Table 1: Notation to refer to the variables and the constraints in our problem, introduced in Sections 2 and 3.3, respectively.

| Symbol | Definition | Explanation |
|---------------------|---|--|
| x^{S_i} | $x^{S_i} = \{x_1^{S_i}, \dots, x_{N_T}^{S_i}\}$ | Time-varying variables pertaining to sensor S_i |
| x^{B_i} | $x^{B_i} = \{x_1^{B_i}, \dots, x_{N_T}^{B_i}\}$ | Time-varying variables pertaining to body segment B_i |
| x^i | $x^i = \{x^{B_i}, x^{S_i}\}$ | Time-varying variables pertaining to sensor S_i and body segment B_i |
| x_t | $x_t = \{x_t^{S_1}, \dots, x_t^{S_{N_S}}, x_t^{B_1}, \dots, x_t^{B_{N_S}}\}$ | Time-varying variables pertaining to time t |
| x | $x = \{x_1, \dots, x_{N_T}\}$ | All time-varying variables |
| θ | $\theta = \{\theta^{S_1}, \dots, \theta^{S_{N_S}}\}$ | Static parameters |
| $c^i(x^i, x^{i+1})$ | $c^i(x^i, x^{i+1}) = \{c_1^i(x_1^i, x_1^{i+1}), \dots, c_{N_T}^i(x_{N_T}^i, x_{N_T}^{i+1})\}$ | Biomechanical constraints for joint i at time $t = 1, \dots, N_T$ |
| $c_t(x_t)$ | $c_t(x_t) = \{c_t^1(x_t^1, x_t^2), \dots, c_t^{N_S-1}(x_t^{N_S-1}, x_t^{N_S})\}$ | Biomechanical constraints at time t |
| $c(x)$ | $c(x) = \{c_t(x_1), \dots, c_t(x_{N_T})\}$ | All biomechanical constraints |

(MAP) estimate,

$$\begin{aligned} & \underset{x, \theta}{\text{maximize}} \quad p(x, \theta \mid y) \\ & \text{subj. to} \quad c(x) = 0, \end{aligned} \tag{1}$$

where $c(x)$ represents the equality constraints. In this work we consider N_S sensors placed on N_S body segments, where sensor S_i is placed on body segment B_i . The time-varying variables x consist of variables both related to sensors (e.g. the pose of the sensor) and to body segments (the pose of the body segment), i.e. $x_t = \{x_t^{S_1}, \dots, x_t^{S_{N_S}}, x_t^{B_1}, \dots, x_t^{B_{N_S}}\}$. To refer to the time-varying variables for sensor S_i , we use the notation $x^{S_i} = \{x_1^{S_i}, \dots, x_{N_T}^{S_i}\}$, while $x^{B_i} = \{x_1^{B_i}, \dots, x_{N_T}^{B_i}\}$ denotes time-varying variables for body segment B_i . The set of time-varying variables pertaining to sensor S_i and segment B_i is denoted $x^i = \{x^{B_i}, x^{S_i}\}$. The static parameters are given by $\theta = \{\theta^{S_1}, \dots, \theta^{S_{N_S}}\}$. This notation will be used throughout this work and is summarized in Table 1.

Using the Markov property of the time-varying variables and the fact that the logarithm is a monotonic function, we can rewrite (1) as

$$\begin{aligned} & \underset{x, \theta}{\text{minimize}} \quad - \underbrace{\sum_{t=2}^{N_T} \sum_{i=1}^{N_S} \log p(x_t^{S_i} \mid x_{t-1}^{S_i}, \theta^{S_i}, y_t^{S_i})}_{\text{dynamics of the state } x_t^{S_i}} \\ & \quad - \underbrace{\sum_{t=1}^{N_T} \sum_{i=1}^{N_S} \log p(x_t^{B_i} \mid x_t^{S_i})}_{\text{placement of sensor } S_i \text{ on body segment } B_i} \\ & \quad - \underbrace{\sum_{i=1}^{N_S} \log p(x_1^{S_i} \mid y_1^{S_i}) - \sum_{i=1}^{N_S} \log p(\theta^{S_i})}_{\text{prior}} \\ & \text{subj. to } c(x) = 0. \end{aligned} \tag{2}$$

The constraints $c(x)$ represent the connection between the body segments at the joint locations. The cost function consists of terms related to a dynamic model for the time-varying states $x_t^{S_i}$, a model regarding the placement of the sensors on the body segments and a prior on the initial states $x_1^{S_i}$ and the constant parameters θ^{S_i} for $i = 1, \dots, N_S$.

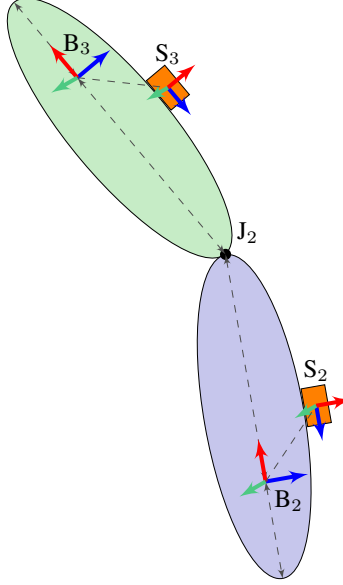


Figure 2: Connection of two body segments with a sensor attached to each of them.

Table 2: Summary of the body segments, sensors and joints used in the model.

| Body segment | Sensor | Joint | Connecting body segments |
|----------------------------------|----------------|------------------------------|---|
| B ₁ : Right foot | S ₁ | J ₁ : Right ankle | B ₁ \Leftrightarrow B ₂ |
| B ₂ : Right lower leg | S ₂ | J ₂ : Right knee | B ₂ \Leftrightarrow B ₃ |
| B ₃ : Right upper leg | S ₃ | J ₃ : Right hip | B ₃ \Leftrightarrow B ₄ |
| B ₄ : Pelvis | S ₄ | J ₄ : Left hip | B ₄ \Leftrightarrow B ₅ |
| B ₅ : Left upper leg | S ₅ | J ₅ : Left knee | B ₅ \Leftrightarrow B ₆ |
| B ₆ : Left lower leg | S ₆ | J ₆ : Left ankle | B ₆ \Leftrightarrow B ₇ |
| B ₇ : Left foot | S ₇ | | |

3 Model

To estimate the relative pose of the lower body, we assume that 7 sensors are placed on different body segments. For notational simplicity, we assume that sensor S_i is attached to body segment B_i . The body segments are connected at the joint locations. Figure 2 illustrates two body segments, which can be thought of as the upper leg (B_3) and the lower leg (B_2). A sensor is attached to each body segment and the body segments are connected at the joint J_2 (the knee). Estimating the relative pose of the body amounts to estimating the position and orientation of the sensors and the body segments using the sensor measurements and the information that the body segments are connected. The variables considered optimization problem (2) are given by:

- The time-varying variables $x_t^{S_i}$, consisting of the 3D position, velocity and orientation of sensor S_i at time t . Furthermore, for one of the sensors S_i , the variables $x_t^{S_i}$ also include variables to estimate its mean acceleration at time t .
- The time-varying variables $x_t^{B_i}$ consisting of the 3D position and orientation of body segment B_i at time t .
- The constant variables θ^{S_i} consisting of the gyroscope bias $b_{\omega}^{S_i} \in \mathbb{R}^3$ of sensor S_i .

Hence, the variables in the optimization problem are $x \in \mathbb{R}^{(15N_s+3)N_T}$ and $\theta \in \mathbb{R}^{3N_s}$, where it is assumed that the orientation variables are encoded using a three-dimensional vector, see e.g. [13, 14, 15]. In the remainder of this section, we discuss the *structure* of the cost functions and of the constraints in (2).

This structure will be exploited in our message passing algorithm. For an alternative and more explicit formulation of the problem, we refer to [5].

3.1 Dynamics of the state $x_t^{S_i}$

The dynamics in (2) expresses the position, velocity and orientation of each sensor S_i in terms of their values at the time instance $t-1$ and in terms of the constant variables θ^{S_i} . The *change* in position, velocity and orientation of sensor S_i is modeled in terms of the acceleration and angular velocity measured by sensor S_i . The mean acceleration is modeled in terms of $x_{t-1}^{S_i}$, θ^{S_i} and the accelerometer measurements. For more details on the acceleration model we refer to [5]. The dynamics of the state $x_t^{S_i}$ can hence be expressed as in (2).

3.2 Placement of the sensors on the body segments

As shown in Figure 2, sensor S_i is assumed to be attached to the body segment B_i . We assume that the relative position and orientation of the sensors on the body segments is known from calibration. At each time instance, the position and orientation of sensor S_i can therefore be expressed in terms of the position and orientation of the body segment B_i . Ideally, this can be incorporated using equality constraints in (2). However, it is physically impossible to place the sensor directly on the bone. Hence, it has to be placed on the soft tissue and the sensor will inevitably move slightly with respect to the bone. To allow for small random movements of the sensor, we incorporate the knowledge about the placement of the sensors on the body segments in the cost function.

3.3 Biomechanical constraints

The constraints $c(x)$ in the optimization problem (2) enforce the body segments to be connected at the joint locations at all times. Hence, for joint J_i , they model the position and the orientation of body segment B_i in terms of the position and the orientation of body segment B_{i+1} for $i = 1, \dots, N_S - 1$. Here, the ordering of the indices of the joints and the body segments is assumed to be as in Table 2. Note that we assume that the length of the body segments is known either from calibration or from a biomechanical model.

Each joint J_i results in a constraint $c_t^i \in \mathbb{R}^3$ at time t . The set of constraints at time t is given by $c_t(x_t) = \{c_t^1(x_t^1, x_t^2), \dots, c_t^{N_S-1}(x_t^{N_S-1}, x_t^{N_S})\}$ and the set of constraints for joint J_i is given by $c^i(x^i, x^{i+1}) = \{c_1^i(x_1^i, x_1^{i+1}), \dots, c_{N_T}^i(x_{N_T}^i, x_{N_T}^{i+1})\}$. The complete set of biomechanical constraints is given by $c(x) = \{c_t(x_1), \dots, c_t(x_{N_T})\}$. This notation is summarized in Table 1. Note that we explicitly indicate which states are involved in the constraints using the ordering of body segments and joints in Table 2.

4 Problem Reformulation Enabling Structure Exploitation

In this section we focus on reordering the problem (2) in two different ways. In Section 4.1, we reorder the problem based on the time indices $t = 1, \dots, N_T$. In Section 4.2, we reorder the problem based on sensor and body segment indices $i = 1, \dots, N_S$. The inertial motion capture problem can be solved iteratively using SQP, where at each iteration k we solve a quadratic approximation of (2). Hence, in each of the sections below, we also introduce an explicit formulation of the quadratic approximation that needs to be solved, where the reordering will allow us to exploit the structure inherent in the problem.

4.1 Reordering based on time

The objective function in (2) can be reordered based on time resulting in

$$\begin{aligned}
& \underset{x, \theta}{\text{minimize}} - \sum_{i=1}^{N_S} \left(\log p(x_1^{B_i} | x_1^{S_i}) + \log p(x_1^{S_i} | y_1^{S_i}) + \frac{1}{N_T} \log p(\theta^{S_i}) \right) \\
& \quad - \sum_{t=2}^{N_T} \sum_{i=1}^{N_S} \left(\log p(x_t^{S_i} | x_{t-1}^{S_i}, \theta^{S_i}) + \log p(x_t^{B_i} | x_t^{S_i}) + \frac{1}{N_T} \log p(\theta^{S_i}) \right) \\
& \text{subj. to } c(x) = 0.
\end{aligned} \tag{3}$$

Let $f_1(x_1, \theta)$ and $f_t(x_t, x_{t-1}, \theta)$ for $t = 2, \dots, N_T$ correspond to different terms in the cost function of (3). We can then rewrite (3) more compactly as

$$\begin{aligned} \underset{x, \theta}{\text{minimize}} \quad & f_1(x_1, \theta) + \sum_{t=2}^{N_T} f_t(x_t, x_{t-1}, \theta) \\ \text{subj. to} \quad & c_t(x_t) = 0, \quad t = 1, \dots, N_T, \end{aligned} \quad (4)$$

where we use the notation $c_t(x_t)$ to denote the biomechanical constraints at time t as introduced in Table 1. It is beneficial to equivalently reformulate this problem as

$$\begin{aligned} \underset{x, \bar{\theta}}{\text{minimize}} \quad & f_1(x_1, \bar{\theta}_1) + \sum_{t=2}^{N_T} f_t(x_t, x_{t-1}, \bar{\theta}_t) \\ \text{subj. to} \quad & c_t(x_t) = 0, \quad t = 1, \dots, N_T, \\ & \bar{\theta}_t = \bar{\theta}_{t+1}, \quad t = 1, \dots, N_T - 1, \end{aligned} \quad (5)$$

where $\bar{\theta} = \{\bar{\theta}_1, \dots, \bar{\theta}_{N_T}\}$. This formulation models the *constant* variables θ in terms of *time-varying* variables $\bar{\theta}_t$. Inclusion of the additional equality constraints in (5) ensures that $\bar{\theta}_t$ will be equal for all t and makes the formulations (4) and (5) equivalent.

The reordered problem (5) enjoys a desirable structure that can be exploited. It can be solved iteratively using SQP, where at each iteration k we solve the quadratic approximation

$$\begin{aligned} \underset{\Delta x, \Delta \bar{\theta}}{\text{minimize}} \quad & \frac{1}{2} \begin{bmatrix} \Delta x \\ \Delta \bar{\theta} \end{bmatrix}^\top H(x^{(k)}, \bar{\theta}^{(k)}) \begin{bmatrix} \Delta x \\ \Delta \bar{\theta} \end{bmatrix} + \left(J_f(x^{(k)}, \bar{\theta}^{(k)}) \right)^\top \begin{bmatrix} \Delta x \\ \Delta \bar{\theta} \end{bmatrix} \\ \text{subj. to} \quad & c_t(x_t^{(k)}) + \left(J_{c_t}(x_t^{(k)}) \right)^\top \Delta x_t = 0, \quad t = 1, \dots, N_T, \\ & \Delta \bar{\theta}_t - \Delta \bar{\theta}_{t+1} = 0, \quad t = 1, \dots, N_T - 1, \end{aligned} \quad (6)$$

to compute a step, $[\Delta x^\top \quad \Delta \bar{\theta}^\top]^\top$. This step will be used to update the estimates of the variables x and $\bar{\theta}$. The Jacobians of the objective function and of the constraints are given by

$$J_f(x, \bar{\theta}) = \nabla_{x, \bar{\theta}} f_1(x_1, \bar{\theta}_1) + \sum_{t=2}^{N_T} \nabla_{x, \bar{\theta}} f_t(x_t, x_{t-1}, \bar{\theta}_t), \quad (7a)$$

$$J_{c_t}(x_t) = \nabla_{x_t} c_t(x_t). \quad (7b)$$

For the Hessian of the objective function we use a Gauss-Newton approximation as

$$H(x, \bar{\theta}) \approx \nabla_{x, \bar{\theta}} f_1(x_1, \bar{\theta}_1) \nabla_{x, \bar{\theta}} f_1(x_1, \bar{\theta}_1)^\top + \sum_{t=2}^{N_T} \nabla_{x, \bar{\theta}} f_t(x_t, x_{t-1}, \bar{\theta}_t) \nabla_{x, \bar{\theta}} f_t(x_t, x_{t-1}, \bar{\theta}_t)^\top. \quad (8)$$

If we choose the ordering of variables as $(\Delta x_1, \Delta \bar{\theta}_1, \Delta x_2, \Delta \bar{\theta}_2, \dots, \Delta x_{N_T}, \Delta \bar{\theta}_{N_T})$, the Hessian $H(x, \bar{\theta})$ takes a special form as illustrated in Figure 3. In this case it is possible to find matrices H_t and h_t and write the problem in (6) equivalently as

$$\begin{aligned} \underset{\Delta x, \Delta \bar{\theta}}{\text{minimize}} \quad & \sum_{t=1}^{N_T-1} \left(\frac{1}{2} \begin{bmatrix} \Delta x_t \\ \Delta \bar{\theta}_t \\ \Delta x_{t+1} \\ \Delta \bar{\theta}_{t+1} \end{bmatrix}^\top H_t \begin{bmatrix} \Delta x_t \\ \Delta \bar{\theta}_t \\ \Delta x_{t+1} \\ \Delta \bar{\theta}_{t+1} \end{bmatrix} + \begin{bmatrix} \Delta x_t \\ \Delta \bar{\theta}_t \\ \Delta x_{t+1} \\ \Delta \bar{\theta}_{t+1} \end{bmatrix}^\top h_t \right) \\ \text{subj. to} \quad & c_t(x_t^{(k)}) + \left(J_{c_t}(x_t^{(k)}) \right)^\top \Delta x_t = 0, \quad t = 1, \dots, N_T, \\ & \Delta \bar{\theta}_t - \Delta \bar{\theta}_{t+1} = 0, \quad t = 1, \dots, N_T - 1. \end{aligned} \quad (9)$$

This time-ordered equivalent formulation of the problem (2) enjoys a special structure which allows us solve it efficiently using message passing. Before introducing this approach, we will first reorder the problem (2) in a second way, based on sensors and body segments.

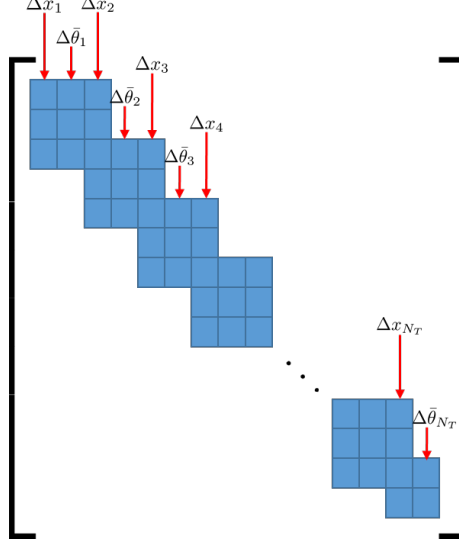


Figure 3: Form of the Hessian $H(x, \bar{\theta})$ for the quadratic approximation (6) with the time-ordered variables as described in Section 4.1. The blue blocks indicate the non-zero terms in the Hessian. For clarity, we indicate which variables are associated with which blocks.

4.2 Reordering based on sensors and body segments

The problem (2) can also be rearranged or reordered based on sensors and body segments. Here, we group the terms in the cost function related to sensor S_i and body segment B_i for $i = 1, \dots, N_S$, resulting in

$$\begin{aligned} & \underset{x, \theta}{\text{minimize}} - \sum_{i=1}^{N_S} \left(\log p(x_1^{S_i} | y_1^{S_i}) + \sum_{t=2}^{N_T} \log p(x_t^{S_i} | x_{t-1}^{S_i}, \theta^{S_i}) + \sum_{t=1}^{N_T} \log p(x_t^{B_i} | x_t^{S_i}) + \log p(\theta^{S_i}) \right) \\ & \text{subj. to } c(x) = 0. \end{aligned} \quad (10)$$

Letting each term in the cost function be denoted by $g^i(x^i, \theta^{S_i})$, we can write (10) compactly as

$$\begin{aligned} & \underset{x, \theta}{\text{minimize}} \quad \sum_{i=1}^{N_S} g^i(x^i, \theta^{S_i}) \\ & \text{subj. to} \quad c^i(x^i, x^{i+1}) = 0, \quad i = 1, \dots, N_S - 1, \end{aligned} \quad (11)$$

where the constraints are grouped per joint. Note again that x^i and $c^i(x^i, x^{i+1})$ are defined in Table 1.

Analogously to the development in Section 4.1, solving the problem in (11) using SQP amounts to solving

$$\begin{aligned} & \underset{\Delta x, \Delta \theta}{\text{minimize}} \quad \frac{1}{2} \begin{bmatrix} \Delta x \\ \Delta \theta \end{bmatrix}^\top \bar{H}(x^{(k)}, \theta^{(k)}) \begin{bmatrix} \Delta x \\ \Delta \theta \end{bmatrix} + \left(J_g(x^{(k)}, \theta^{(k)}) \right)^\top \begin{bmatrix} \Delta x \\ \Delta \theta \end{bmatrix} \\ & \text{subj. to} \quad c^i(x^{i,(k)}, x^{i+1,(k)}) + \left(J_{c^i}(x^{i,(k)}, x^{i+1,(k)}) \right)^\top [\Delta x^i \Delta x^{i+1}] = 0, \quad i = 1, \dots, N_S - 1, \end{aligned} \quad (12)$$

at each iteration, where

$$J_g(x^{(k)}, \theta^{(k)}) = \sum_{i=1}^{N_S} \nabla_{x, \theta} g^i(x^i, \theta^{S_i}), \quad (13a)$$

$$J_{c^i}(x^i, x^{i+1}) = \nabla_{x^i, x^{i+1}} c^i(x^i, x^{i+1}). \quad (13b)$$

The Hessian of the objective function of this problem is again based on a Gauss-Newton approximation,

$$\bar{H}(x, \theta) \approx \sum_{i=1}^{N_S} \nabla_{x, \theta} g^i(x^i, \theta^{S_i}) \nabla_{x, \theta} g^i(x^i, \theta^{S_i})^\top. \quad (14)$$

If we choose the ordering of variables as $(x^1, \theta^{S_1}, x^2, \theta^{S_2}, \dots, x^{N_S}, \theta^{S_{N_S}})$, the Hessian becomes block-diagonal with each block corresponding to sensor S_i and body segment B_i . This then enables us to write the problem in (12) as

$$\begin{aligned} \underset{\Delta x, \Delta \theta}{\text{minimize}} \quad & \sum_{i=1}^{N_S-1} \left(\frac{1}{2} \begin{bmatrix} \Delta x^i \\ \Delta \theta^{S_i} \\ \Delta x^{i+1} \\ \Delta \theta^{S_{i+1}} \end{bmatrix}^\top \bar{H}^i \begin{bmatrix} \Delta x^i \\ \Delta \theta^{S_i} \\ \Delta x^{i+1} \\ \Delta \theta^{S_{i+1}} \end{bmatrix} + \begin{bmatrix} \Delta x^i \\ \Delta \theta^{S_i} \\ \Delta x^{i+1} \\ \Delta \theta^{S_{i+1}} \end{bmatrix}^\top \bar{h}^i \right) \\ \text{subj. to} \quad & c^i(x^{i,(k)}, x^{i+1,(k)}) + \left(J_{c^i}(x^{i,(k)}, x^{i+1,(k)}) \right)^\top \begin{bmatrix} \Delta x^i \\ \Delta x^{i+1} \end{bmatrix} = 0, \quad i = 1, \dots, N_S - 1, \end{aligned} \quad (15)$$

through consistent choices of matrices \bar{H}^i and vectors \bar{h}^i . The problem formulation (15) again enjoys a special structure which allows us to solve it efficiently using message passing. Next we briefly review this approach.

5 Tree Structure in Coupled Problems and Message Passing

Consider the following coupled optimization problem

$$\underset{z}{\text{minimize}} \quad f_1(z) + f_2(z) + \dots + f_{N_C}(z), \quad (16)$$

where $z \in \mathbb{R}^{n_z}$ and $f_a : \mathbb{R}^{n_z} \rightarrow \mathbb{R}$ for $a = 1, \dots, N_C$. This problem can be seen as a combination of N_C subproblems, each of which is defined by a term in the cost function and depends only on a few elements of z . Note that f_a can include indicator functions on constraints. Hence, the problem formulations of the inertial motion capture problem (5), (9) for the time ordering and (11), (15) for the sensor and body segment ordering, are of the form (16).

Let us denote the ordered set of indices of z that each subproblem a depends on by C_a . We can then equivalently rewrite (16) as

$$\underset{z}{\text{minimize}} \quad \bar{f}_1(z_{C_1}) + \dots + \bar{f}_{N_C}(z_{N_C}), \quad (17)$$

where z_{C_a} is a $|C_a|$ -dimensional vector that contains the elements of z indexed by C_a , with $|C_a|$ denoting the number of elements in the set C_a . Also the functions $\bar{f}_a : \mathbb{R}^{|C_a|} \rightarrow \mathbb{R}$ are lower dimensional descriptions of f_a s such that $f_a(z) = \bar{f}_a(z_{C_a})$ for all z and $a = 1, \dots, N_C$. It is possible to describe the coupling structure of the problem graphically using undirected graphs. Particularly, let us define the *sparsity graph* of the problem as a graph $G_s(V_s, \mathcal{E}_s)$ with the vertex set $V_s = \{1, \dots, n_z\}$ and $(a, b) \in \mathcal{E}_s$ if and only if variables z_a and z_b appear in the same subproblem. Let us assume that each C_a for $a = 1, \dots, N_C$, be a *clique* of this graph, where a clique is a maximal subset of V_s that induces a complete subgraph on G_s . This in turn means that no clique is contained in another clique [16]. Assume furthermore that there exists a tree defined on \mathbf{C}_{G_s} such that for every $C_a, C_b \in \mathbf{C}_{G_s}$ where $a \neq b$, $C_a \cap C_b$ is contained in all the cliques in the path connecting the two cliques in the tree. This property is called the clique intersection property [16]. Graphs with this property have an *inherent tree structure* and can be represented using a *clique tree*.

Let us assume that the sparsity graph of the problem (17) has an inherent tree structure. The problem can then be solved distributedly using a message passing algorithm that utilizes the clique tree as its computational graph. This means that the nodes $V_c = \{1, \dots, N_C\}$ act as computational agents that communicate or collaborate with their neighbors defined by the edge set \mathcal{E}_c . The message-passing algorithm solves (17) by performing an upward-downward pass through the clique tree, see e.g., [8, 17] and references therein. The upward pass starts from the agents at the leaves of the tree, i.e., all agents $a \in \text{leaves}(T)$, where every such agent computes and communicates the message

$$m_{a \text{ par}(a)}(z_{A_{a \text{ par}(a)}}) = \min_{z_{C_a \setminus A_{a \text{ par}(a)}}} \{ \bar{f}_a(z_{C_a}) \}, \quad (18)$$

to its parent, denoted by $\text{par}(a)$. Here $A_{ab} = C_a \cap C_b$ is the so-called separator set of agents a and b . Then every agent a that has received all messages from its children, computes and communicates the message

$$m_{a \text{ par}(a)}(z_{A_{a \text{ par}(a)}}) = \min_{z_{C_a \setminus A_{a \text{ par}(a)}}} \left\{ \bar{f}_a(z_{C_a}) + \sum_{b \in \text{ch}(a)} m_{ba}(z_{A_{ba}}) \right\}, \quad (19)$$

Algorithm 1 Distributed Optimization Using Message Passing

```
1: Given a sparsity graph  $G_s$  of an optimization problem
2:   extract its cliques and a clique tree over the cliques;
3:   assign each subproblem to one and only one of the agents.
4: Set  $\text{agents} = \{1, \dots, N_C\} \setminus r$  and  $\text{elim} = \{\}$ .
5: Perform the upward pass as
6: while  $|\text{agents}| \neq 0$  do
7:   for  $i \in \text{agents}$  do
8:     if  $\text{ch}(a) \subseteq \text{elim}$  then
9:       This agent computes the message in (19) and communicates it to agent  $\text{par}(a)$ .
10:       $\text{elim} = \text{elim} \cup \{a\}$ .
11:    end if
12:  end for
13:   $\text{agents} = \text{agents} \setminus \text{elim}$ .
14: end while
15: Set  $\text{agents} = \{1, \dots, N_C\}$  and  $\text{elim} = \{\}$ .
16: Perform the downward pass as
17: while  $|\text{agents}| \neq 0$  do
18:   for  $a \in \text{agents}$  do
19:     if  $\text{par}(a) \subseteq \text{elim}$  then
20:       This agent computes optimal solution as in (21) and communicates it to agents  $\text{ch}(a)$ .
21:        $\text{elim} = \text{elim} \cup \{a\}$ .
22:     end if
23:   end for
24:    $\text{agents} = \text{agents} \setminus \text{elim}$ .
25: end while
26: By the end of the downward pass all agents have computed their optimal solutions and the algorithm is
    terminated.
```

with $\text{ch}(a)$ denoting the children of agent a , to its parent. This procedure is continued until we reach the agent, r , at the root. At this point, agent r computes its corresponding optimal solution by solving

$$z_{C_r}^* = \arg \min_{z_{C_r}} \left\{ \bar{f}_r(z_{C_r}) + \sum_{b \in \text{ch}(r)} m_{br}(z_{A_{br}}) \right\}, \quad (20)$$

and initiates the downward pass by communicating this solution to its children. During the downward pass each agent a having received the optimal solution $(z_{A_{\text{par}(a)a}}^*)^{\text{par}(a)}$ from its parent computes its corresponding optimal solution as

$$z_{C_a}^* = \arg \min_{z_{C_a}} \left\{ \bar{f}_a(z_{C_a}) + \sum_{b \in \text{ch}(a)} m_{ba}(z_{A_{ba}}) + \frac{1}{2} \left\| z_{A_{\text{par}(a)a}} - (z_{A_{\text{par}(a)a}}^*)^{\text{par}(a)} \right\|^2 \right\}, \quad (21)$$

and communicates this solution to its children. Once the downward pass is accomplished, all agents have computed their respective optimal solution and the algorithm is terminated. We have summarized this scheme in Algorithm 1.

Remark 1 Notice that within the upward pass all agents that have received messages from their children can compute their messages simultaneously and in parallel. This also holds for the downward pass, as all agents that have received the optimal solution from their parents can compute their respective optimal solution in parallel.

6 Scalable and Distributed Solutions Using Message Passing

We will now rewrite the problem reorderings (9) and (15) such that Algorithm 1 can be used to solve the problem. Let us first reconsider the problem in (9). We can rewrite this problem compactly as

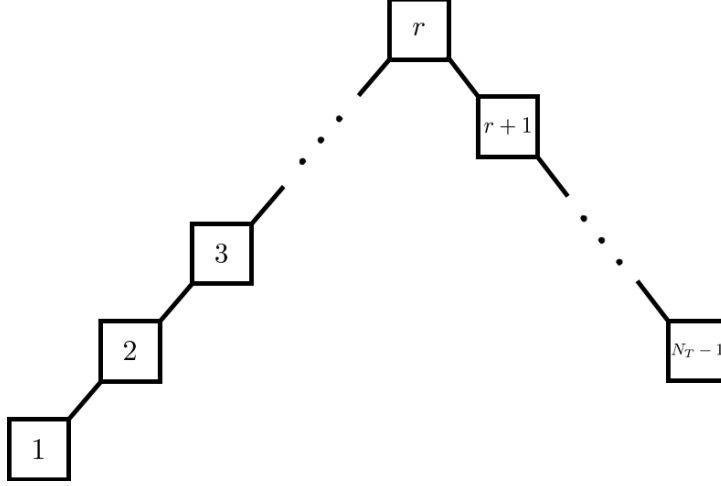


Figure 4: Clique tree corresponding to the sparsity graph of problem (9).

$$\begin{aligned} & \underset{\Delta x, \Delta \bar{\theta}}{\text{minimize}} && \sum_{t=1}^{N_T-1} \bar{f}_t(\Delta x_t, \Delta x_{t+1}, \Delta \bar{\theta}_t, \Delta \bar{\theta}_{t+1}) \end{aligned} \quad (22a)$$

subj. to

$$\left. \begin{aligned} c_t(x_t^{(k)}) + \left(J_{c_t}(x_t^{(k)}) \right)^\top \Delta x_t &= 0 \\ \Delta \bar{\theta}_t - \Delta \bar{\theta}_{t+1} &= 0 \end{aligned} \right\}, t = 1, \dots, r-1, \quad (22b)$$

$$\left. \begin{aligned} c_t(x_t^{(k)}) + \left(J_{c_t}(x_t^{(k)}) \right)^\top \Delta x_t &= 0 \\ \Delta \bar{\theta}_t - \Delta \bar{\theta}_{t+1} &= 0 \end{aligned} \right\}, t = r, \quad (22c)$$

$$\left. \begin{aligned} c_t(x_{t+1}^{(k)}) + \left(J_{c_{t+1}}(x_{t+1}^{(k)}) \right)^\top \Delta x_{t+1} &= 0 \\ \Delta \bar{\theta}_t - \Delta \bar{\theta}_{t+1} &= 0 \end{aligned} \right\}, t = r+1, \dots, N_T-1, \quad (22d)$$

where $r = \lfloor N_T/2 \rfloor$. The sparsity graph of this problem has an inherent tree structure, with $N_T - 1$ cliques. Each clique C_a consists of the variables Δx_a , $\Delta \bar{\theta}_a$, Δx_{a+1} and $\Delta \bar{\theta}_{a+1}$. The clique tree for this problem is illustrated in Figure 4. Consequently, we can use Algorithm 1 for solving this problem. During the upward pass, each agent a sends a message as in (18) and (19) to its parent as a function of the variables it shares with its parents. Hence, if $a < r$ (the agent is on the left side of agent r in Figure 4) the message to its parents is a function of Δx_{a+1} and $\Delta \bar{\theta}_{a+1}$. Equivalently, if $a > r$ (the agent is on the right side of agent r in Figure 4) the message to its parents is a function of Δx_a and $\Delta \bar{\theta}_a$. As a result, each agent except agent r has to factorize a matrix of size $|x_a| + |\bar{\theta}_a|$ plus the number of constraints, which is equal to $6N_S - 3$, as can be seen in (22). The root agent instead has to factorize a matrix of size $2|x_a| + 2|\bar{\theta}_a| + 9N_S - 6$. The computational complexity of Algorithm 1 is dominated by the upward pass since the downward pass does not require a matrix factorization. For details on this, we refer to [8]. Hence, the computational complexity and storage requirements for the resulting algorithm grow linearly with N_T . The reduction in the memory requirements follows from the fact that using Algorithm 1 we have relaxed the need for even forming the problem in (9). The resulting algorithm to solve the problem (22) is summarized in Algorithm 2.

The problem in (15) is also a coupled problem but with $N_S - 1$ subproblems. The clique tree for this problem has the same structure as for (15), where the only differences are in the number of cliques which in this case is $N_S - 1$ and that $r = \lfloor N_S/2 \rfloor$. Each clique C_a consists of the variables Δx^a , $\Delta \theta^{S_a}$, Δx^{a+1} and $\Delta \theta^{S_{a+1}}$. Hence, we can solve the problem in (15) distributedly using Algorithm 1. This can be achieved using a network of computational agents, that can be installed on the body and that collaborate based on the clique tree.

Algorithm 2 Inertial Motion Capture

- 1: Place the sensors on the body, calibrate the system and collect inertial measurements.
 - 2: Initialize x and θ or $\bar{\theta}$ and set $k = 0$.
 - 3: **while** the algorithm has not converged and the solution is not feasible **do**
 - 4: Formulate the quadratic approximation (22) using the time reordering or (15) using the sensor / segment reordering.
 - 5: Use Algorithm 1 to solve the problem formulated in Step 4 and to obtain a step $[\Delta x^\top \ \Delta \bar{\theta}^\top]^\top$ for the time ordered problem or a step $[\Delta x^\top \ \Delta \theta^\top]^\top$ for the sensor / segment ordered problem.
 - 6: Update $x := x + \Delta x$, $\theta := \theta + \Delta \theta$ or $\bar{\theta} := \bar{\theta} + \Delta \bar{\theta}$.
 - 7: Set $k := k + 1$ and check for convergence and feasibility.
 - 8: **end while**
-

Remark 2 *Note that in (22) we have adopted a particular grouping of the constraints. This is to ensure that each of the subproblems is well-posed in terms of their local variables. This was not necessary for the problem in (15).*

Remark 3 *The reason that the clique trees for both problems in (22) and (15) have the same structure is due to the fact that we have focused on the motion capture problem for the lower body. For solving the full body problem we can use the same approach as presented in this paper, since the inherent tree structure will still be present in the problem. However, the clique tree for the corresponding problem will be more complicated than a chain and will correspond to the body formation.*

7 Results and discussion

We consider experimental data from a subject walking around for approximately 37 seconds wearing inertial sensors as shown in Figure 5. We focus on estimating the pose of the lower body using data from 7 sensors attached to the different body segments as described in Section 3. The estimated joint angles from this problem have previously been presented in [5]. In this work, we solve the same optimization problem but reorder the problem to efficiently make use of its structure. Hence, the estimates obtained using Algorithm 2 are equal to the ones presented in [5].

The optimization problem is solved at 10 Hz with $N_T = 373$, leading to a total number of 40284 time-varying variables x and 21 constant variables θ and 6714 constraints.¹ Notice that if the inherent sparsity of the problem would not be exploited, the computational complexity of solving the SQP for the smoothing problem (6) or (12) would grow cubically with the number of sensors and body segments N_S and with the number of time steps N_T . The storage requirements for forming this problem would grow quadratically with N_T and N_S .

To solve the problem in a more scalable manner, we have reordered the variables based on time and formed the problem as in (22), which allows us to solve the problem using Algorithm 2. For each iteration k in Algorithm 2, we then form $N_T - 1$ subproblems. Computing the messages in the upward pass requires each agent except the root agent to factorize a matrix of size 168 since $|x_t| + |\bar{\theta}_t| = 129$ and 39 constraints are involved in the subproblem. The root agent needs to instead factorize a matrix of size 315 since $2|x_t| + 2|\bar{\theta}| = 258$ and 57 constraints are involved in this subproblem instead. Using message passing to solve the problem, it is no longer required to form and store the large problem of size 46998. Instead, it is only required to store one of these subproblems.

We have also solved the problem by reordering the variables based on sensors and segments. The computational benefits of this reordering are much less significant – the problem is split up in 6 subproblems. However, the approach no longer requires collecting all data at a centralized unit, which can be communication intensive, and hence can potentially hamper our ability to have a seamless solution for the motion capture problem. Instead, it allows for decentralized computation of the solution, where the computational power on the sensors can be used to compute solutions to the subproblems.

¹Note that the inertial sensors themselves are sampled at a much higher rate but strapdown integration [18, 19] is used to capture the high frequency signals, allowing for a lower update frequency of the optimization problem.



Figure 5: Experimental setup where the human body is equipped with inertial sensors (orange boxes) on different body segments. High-accuracy reference measurements were obtained using an optical tracking system to validate the estimated joint angles. To this end, triangles with optical markers were placed on a number of sensors.

8 Conclusions and future work

In this work, we have introduced a method to exploit the structure inherent in the inertial motion capture problem. The method allows for a scalable solution where small subproblems for each time step are formed and hence longer data sets can be processed. The approach is successfully applied to experimental data to estimate the pose of the lower body. It also opens up for the possibility of distributedly solving the problem by making use of the computational resources of each of the sensors. The structure that we exploit in this work is not unique to the motion capture problem. We believe that the message passing algorithm can be applied to a large number of other problems appearing in signal processing and estimation, e.g., in large-scale signal processing and estimation application. This is because these problems commonly enjoy desirable sparsity structures arising from physical and / or dynamic properties in the problem, as we saw for the inertial motion capture problem in this work.

Acknowledgment

This work is supported by CADICS, a Linnaeus Center and by the project Probabilistic modeling of dynamical systems (Contract number: 621-2013-5524), both funded by the Swedish Research Council (VR) and by the Swedish Department of Education within the ELLIIT project.

References

- [1] Xsens Technologies B. V. <http://www.xsens.com>, Accessed on March 5, 2016.
- [2] D. Roetenberg, H. J. Luinge, and P. Slycke. Xsens MVN: Full 6DOF human motion tracking using miniature inertial sensors, May 2013.
- [3] D. H. Kang, Y. J. Jung, A. J. Park, and J. W. Kim. Human body motion capture system using magnetic and inertial sensor modules. In *Proceedings of the 5th international universal communication symposium (IUCS)*, Gumi, Korea, October 2011.
- [4] X. Yun and E. R. Bachmann. Design, implementation, and experimental results of a quaternion-based Kalman filter for human body motion tracking. *IEEE Transactions on Robotics*, 22(6): 1216–1227, 2006.

- [5] M. Kok, J. D. Hol, and T. B. Schön. An optimization-based approach to human body motion capture using inertial sensors. In *Proceedings of the 19th World Congress of the International Federation of Automatic Control*, pages 79–85, Cape Town, South Africa, August 2014.
- [6] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research, 2nd edition, 2006.
- [7] C. V. Rao, J. B. Rawlings, and J. H. Lee. Constrained linear state estimation - a moving horizon approach. *Automatica*, 37(10):1619–1628, 2001.
- [8] S. Khoshfetrat Pakazad, A. Hansson, and M. S. Andersen. Distributed primal-dual interior-point methods for solving loosely coupled problems using message passing. *ArXiv e-prints*, June 2015. arXiv:1502.06384.
- [9] D. P. Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena Scientific Belmont, MA, 3rd edition, 1995.
- [10] H. E. Rauch, C. T. Striebel, and F. Tung. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450, 1965.
- [11] V. Cevher, S. Becker, and M. Schmidt. Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics. *IEEE Signal Processing Magazine*, 31(5):32–43, Sept 2014.
- [12] S. J. Wright. *Primal-dual interior-point methods*. Siam, 1997.
- [13] J. L. Crassidis, F. Landis Markley, and Y. Cheng. A survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28, 2007.
- [14] G. Grisetti, R. Kummerle, C. Stachniss, U. Frese, and C. Hertzberg. Hierarchical optimization on manifolds for online 2D and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 273–278, Anchorage, Alaska, 2010.
- [15] J. D. Hol. *Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra-Wideband and GPS*. PhD thesis, Linköping University, Sweden, June 2011.
- [16] J. R. S. Blair and B. W. Peyton. An introduction to chordal graphs and clique trees. In J. A. George, J. R. Gilbert, and J. W-H. Liu, editors, *Graph Theory and Sparse Matrix Computations*, volume 56, pages 1–27. Springer-Verlag, 1994.
- [17] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.
- [18] P. G. Savage. Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms. *Journal of Guidance, Control and Dynamics*, 21(1):19–28, 1998.
- [19] P. G. Savage. Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithms. *Journal of Guidance, Control and Dynamics*, 21(2):208–221, 1998.